
Systeme de Gestion de Base de Données (SGBD)

Le langage SQL, approfondissement

I. Les opérateurs logiques

- **L'opérateur ALL**

Sert à comparer une valeur à toutes les valeurs d'un ensemble retourné par une sous-requête.

👉 *la comparaison est vrai si elle est vrai pour tous les éléments de l'ensemble.*

Exemple: Chercher le salarié ayant un salaire supérieur à tous les salaires du service n°10.

👉 *La comparaison se fait avec tous les éléments (un par un) retournés par la sous-requête⁺.*

I. Les opérateurs logiques

- **L'opérateur ANY**

Sert à comparer une valeur à toute valeur applicable d'un ensemble retourné par une sous-requête.

👉 *la comparaison est vrai si elle est vrai pour au moins un élément de l'ensemble*

Exemple: Chercher s'il y a un(e) secrétaire ayant un salaire supérieur à un ou plusieurs ingénieurs de la société.

👉 *La condition est vrai dès qu'on rencontre une ligne vérifiant la condition.*

I. Les opérateurs logiques

- **L'opérateur EXISTS**

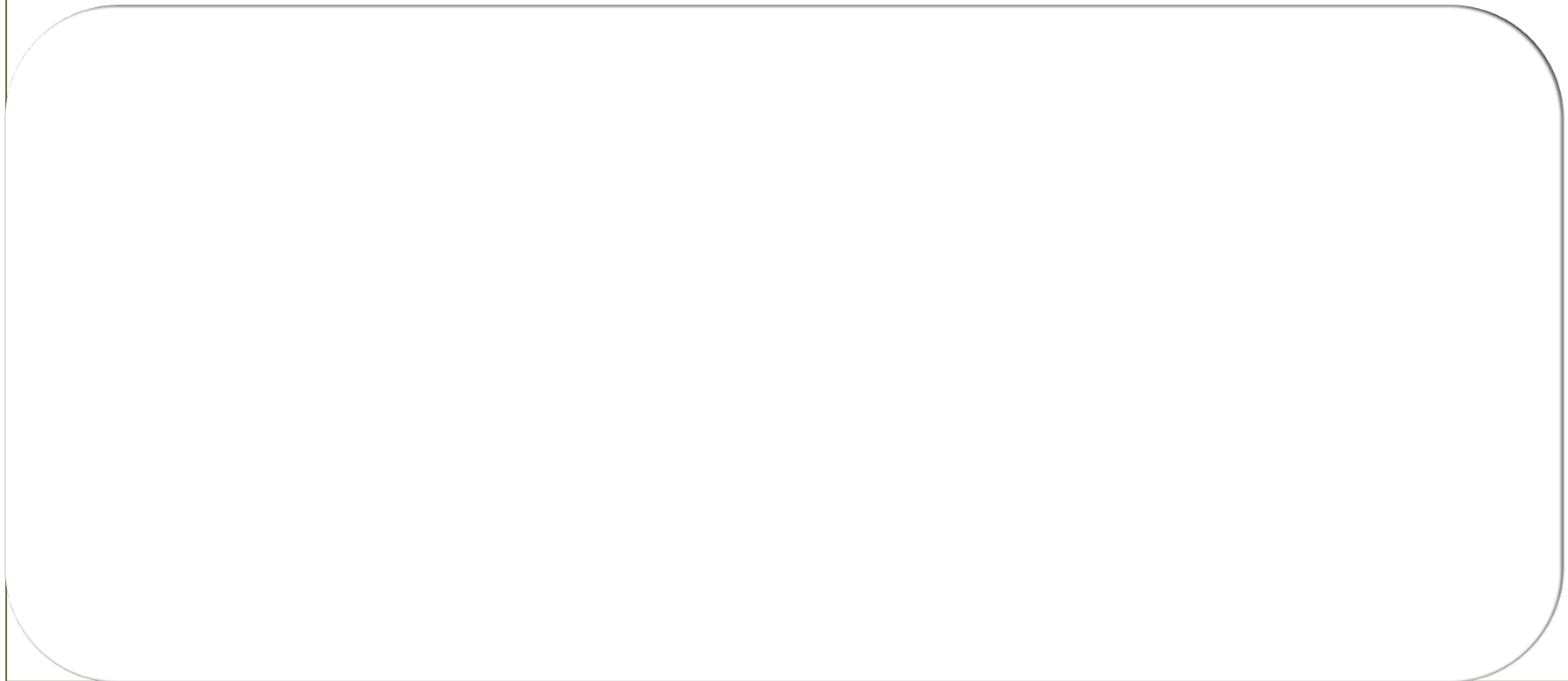
Permet de chercher la présence d'une ligne répondant à certains critères dans une table spécifique: vérifie si une sous-requête retourne au moins une ligne.

Exemple 1: Donner les noms des services ayant au moins un employé embauché après le '01/01/05'.

I. Les opérateurs logiques

- **L'opérateur EXISTS**

Exemple 2: Donner les noms des projets dont aucun employé n'appartient au service FINANCE.



I. Les opérateurs logiques

- **L'opérateur EXISTS**

Exemple 2: Donner les noms des projets dont aucun employé n'appartient au service FINANCE.

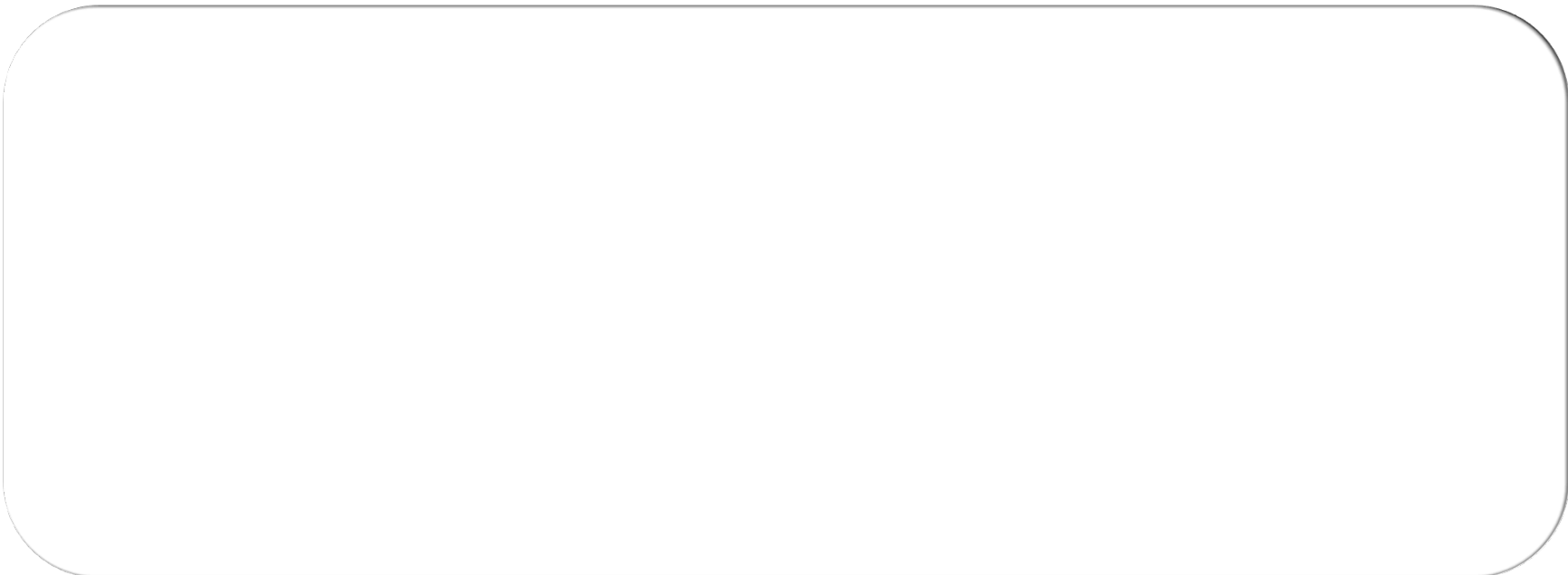
```
SELECT titreproj
FROM prj X
WHERE NOT EXISTS(
    SELECT *
    FROM pers Y, emprj Z
    WHERE X.projno=Z.proj_projno
    AND Y.pno=Z.apers_pno
    AND pserve = (
        SELECT servno
        FROM serv
        WHERE snom='FINANCE' ) );
```

II. Les opérateurs logiques

- **L'opérateur NOT EXISTS**

Permet d'obtenir la division de deux relations.

La division de R par S sur l'attribut B ($R(A,B) \div S(B)$) peut être traduit en SQL par:



II. Les opérateurs logiques

- L'opérateur NOT EXISTS

Exemple: Les athlètes ayant participé au moins à toutes les épreuves.

```
SELECT Athlète  
FROM PARTICIPER A  
WHERE NOT EXISTS (SELECT *
```

```
FROM EPREUVE
```

```
WHERE NOT EXISTS (SELECT *
```

```
FROM PARTICIPER B
```

```
WHERE (A.Athlète=B.Athlète)
```

```
AND(B.Epreuve=EPREUVE.Epreuve)
```

```
));
```

PARTICIPER

Athlète	Epreuve
Dupont	200 m
Dupont	400 m
Dupont	110 m H
Martin	200 m
Martin	400 m
Martin	110 m H
Bertrand	200 m
Bertrand	400 m
Michel	200 m

EPREUVE

Epreuve
200 m
400 m
110 m H



Dupont et Martin

II. Les Requêtes composées

Une requête composée est l'association de plusieurs requêtes par le biais d'un opérateur de combinaison.

Syntaxe:

```
SELECT colonne1, colonne2 FROM TABLE1
WHERE conditions...
opérateur
(SELECT colonne1, colonne2 FROM TABLE2
WHERE conditions...);
```

Opérateurs :

```
UNION,
UNION ALL,
INTERSECT,
MINUS.
```

II. Les Requêtes composées

- **L'opérateur UNION**

permet de combiner les résultats de deux instructions SELECT ou plus, sans retourner des lignes en double. Les colonnes sélectionnées des différentes requêtes doivent être de même nombre, de même type et de même ordre.

Exemple: Donner la liste des salariés travaillant sur des projets comme étant employés ou responsables de projets.

```
SELECT * FROM pers WHERE pno IN (SELECT Epres_pno  
                                FROM emprj);
```

UNION

```
SELECT * FROM pers WHERE pno IN (SELECT Rpers_pno  
                                FROM prj);
```

II. Les Requêtes composées

- **L'opérateur UNION ALL**

Permet de combiner les résultats de deux instructions SELECT y compris les lignes dupliquées. Les règles qui s'appliquent à cet opérateur sont les mêmes que celles qui concernent UNION.

II. Les Requêtes composées

- **L'opérateur INTERSECT**

Permet de combiner les résultats de deux instructions SELECT ou plus, mais ne retourne que les lignes en commun.

Exemple: Donner la liste des salariés qui sont à la fois affectés à des projets et responsables de projets.



III. Les objets des DB

Un objet est une composante stockée dans la base de données.

✓ Liste des objets d'une BD:

Table, contrainte,

Vue, séquence, index, procédure, ...

✓ Visualiser la liste des objets créés par un utilisateur:



III. Les objets des DB : Les Vues

- ✓ Une vue est une table virtuelle constituée par des sous-ensembles de données d'une ou plusieurs tables.
- ✓ Elle est définie par une requête prédéfinie stockée dans la base de données.
- ✓ Elle est exploitée comme une table mais elle ne nécessite pas d'espace physique de stockage.
- ✓ Vue=> fenêtre dynamique, non matérialisée sur les disques, par la quelle un utilisateur accède à la base.

III. Les objets des DB : Les Vues

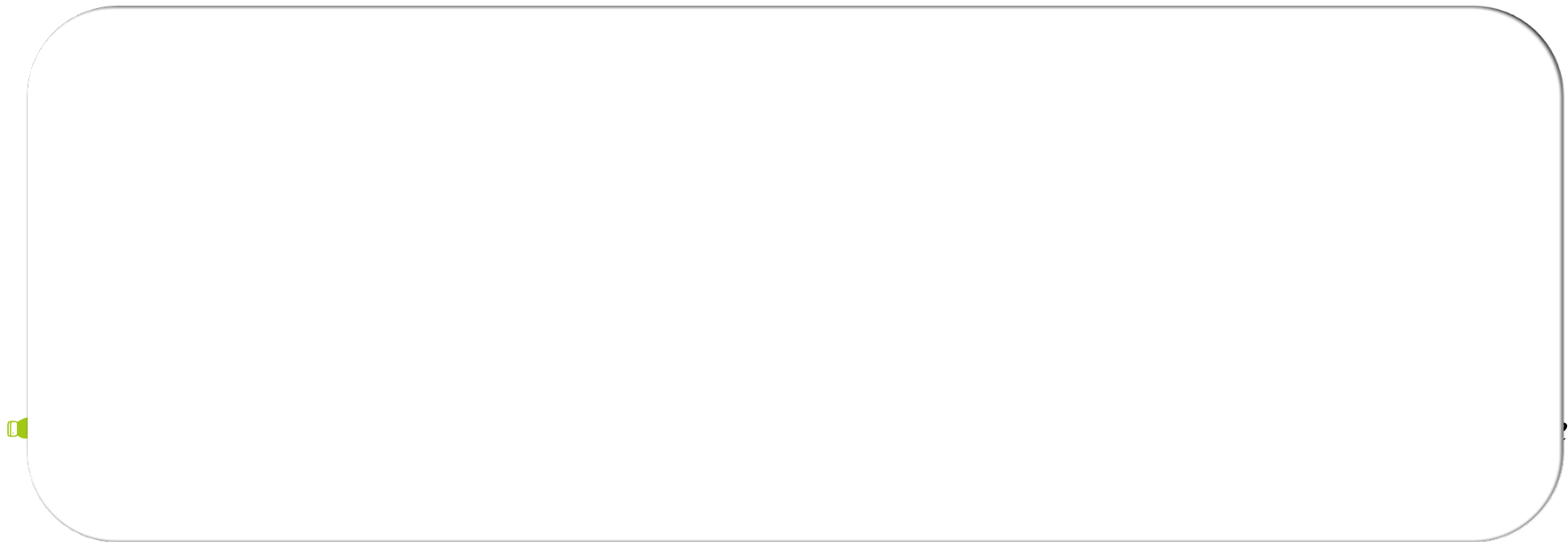
Utilités :

- ✓ Pour des raisons de sécurité : restreindre l'accès aux données.
- ✓ Conserver des données de synthèses afin d'éviter les requêtes très complexes.
- ✓ Permettre une certaine indépendance des données.
- ✓ Présenter des différentes vues pour les mêmes données...

III. Les objets des DB : Les Vues

- **Création d'une vue**
 - **Cas d'une vue simple** (à partir d'une table).

Syntaxe :



III. Les objets des DB : Les Vues

- **Création d'une vue**
 - **Cas d'une vue simple** (à partir d'une table).

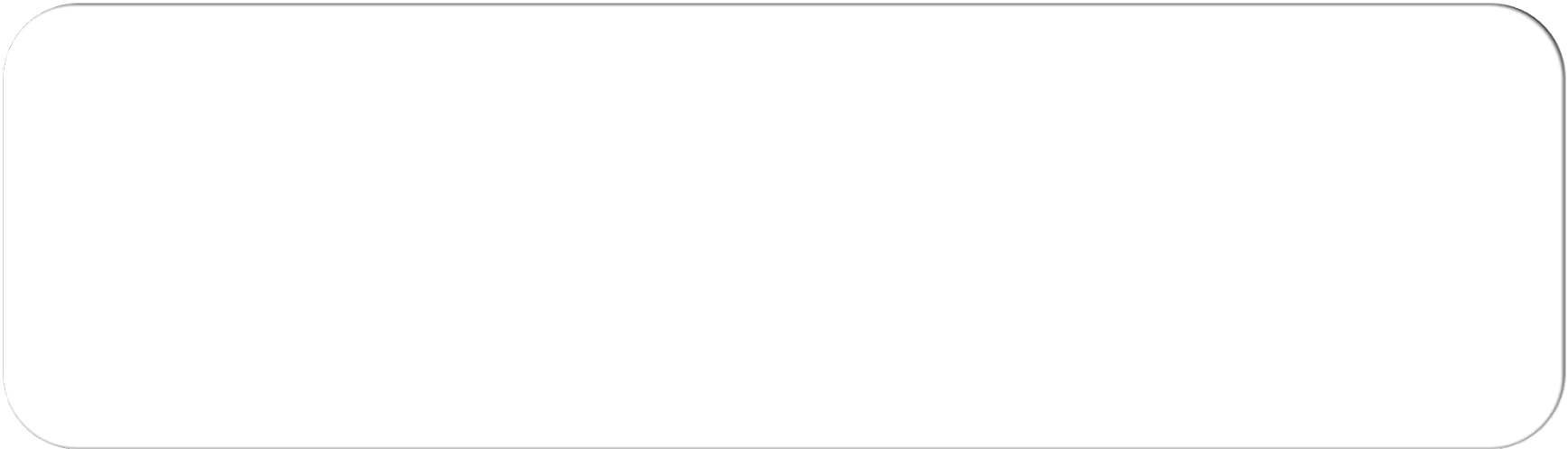
Exemple: Créer une vue des noms, postes et salaires des employés du service n°10.

- ☞ *On peut renommer les noms des colonnes dans la vue en utilisant les alias dans la sous-requête.*

III. Les objets des DB : Les Vues

- **Création d'une vue**
 - **Cas d'une vue complexe** (à partir de plusieurs tables).

Syntaxe :

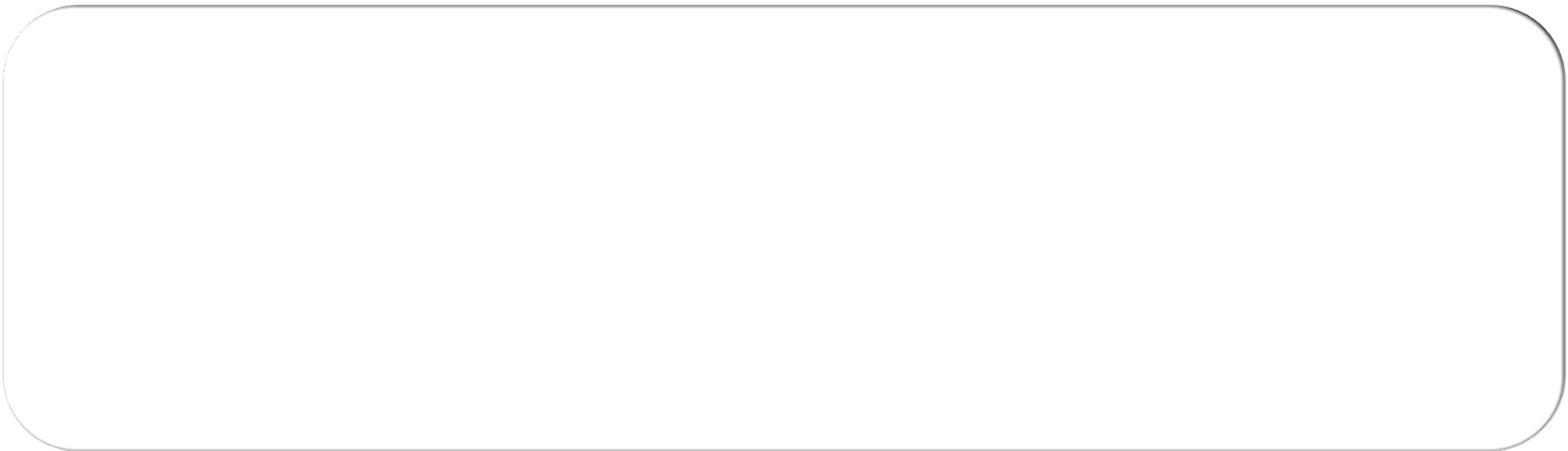


👉 *Les vues complexes sont souvent utilisées pour la synthèse des données.*

III. Les objets des DB : Les Vues

- **Création d'une vue**
 - **Cas d'une vue complexe** (à partir de plusieurs tables).

Exemple : Créer une vue qui contient, pour chaque service, le nom du service, le nombre d'employés et le salaire moyen dans ce service.



III. Les objets des DB : Les Vues

- **Suppression d'une vue**

La suppression d'une vue s'effectue de la même façon qu'une table:

```
Drop nom_vue;
```

Remarques:

Les opérations de LMD (INSERT, UPDATE, DELETE) sont aussi permises sur les vues, par contre les conditions suivantes doivent être vérifiées :

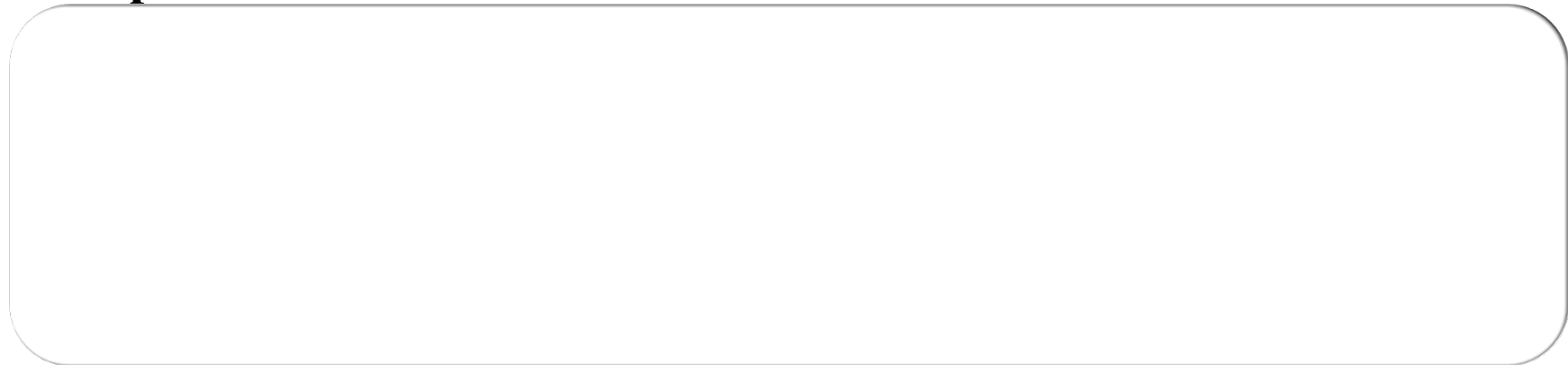
l'exemple précédent.

III. Les objets des DB : Les Vues

- **L'option WITH CHECK OPTION**

C'est une option de l'instruction CREATE qui peut être ajoutée afin que toutes les instructions UPDATE et INSERT répondent aux conditions de la définition de la vue.

Exemple:



👉 *Dans ce cas, l'option WITH CHECK OPTION refuse toute valeur NULL dans la colonne « commission »*